



Répartir les attractions (Split the attractions)

Il y a n attractions à Bakou, numérotées de 0 à $n - 1$. Il y a également m routes à double sens, numérotées de 0 à $m - 1$. Chaque route relie deux attractions distinctes. Il est possible de se déplacer entre chaque paire d'attractions en passant par les routes.

Fatima a prévu de visiter toutes les attractions en trois jours. Elle a déjà décidé qu'elle voulait visiter a attractions le premier jour, b attractions le deuxième jour, et c attractions le troisième jour. Elle va donc partitionner les n attractions en trois ensembles A , B et C , de tailles respectives a , b et c . Chaque attraction appartiendra à exactement un ensemble, donc $a + b + c = n$.

Fatima aimerait choisir les trois ensembles A , B et C de telle sorte qu'**au moins deux** des trois ensembles soient **connexes**. Un ensemble S d'attractions est dit connexe s'il est possible de se déplacer entre chaque paire d'attractions de S en passant par les routes et sans passer par des attractions qui ne sont pas dans S . Une partition des attractions en trois ensembles A , B et C est dite **valide** si elle satisfait les conditions décrites ci-dessus.

Aidez Fatima à trouvez une partition valide des attractions (étant donnés a , b et c), ou déterminez s'il n'en existe aucune. S'il existe plusieurs partitions valides, vous pouvez choisir n'importe laquelle d'entre elles.

Détails d'implémentation

Vous devez implémenter la fonction suivante :

```
int[] find_split(int n, int a, int b, int c, int[] p, int[] q)
```

- n : le nombre d'attractions.
- a , b et c : les tailles voulues pour les ensembles A , B et C .
- p et q : tableaux de taille m , contenant les extrémités des routes. Pour chaque i ($0 \leq i \leq m - 1$), $p[i]$ et $q[i]$ sont les deux attractions reliées par la route i .
- Cette fonction doit renvoyer un tableau de taille n . Appelons ce tableau s . S'il n'y a pas de partition valide, s doit contenir n zéros. Sinon, pour $0 \leq i \leq n - 1$, $s[i]$ doit être égal à 1, 2 ou 3 pour indiquer que l'attraction i fait partie de l'ensemble

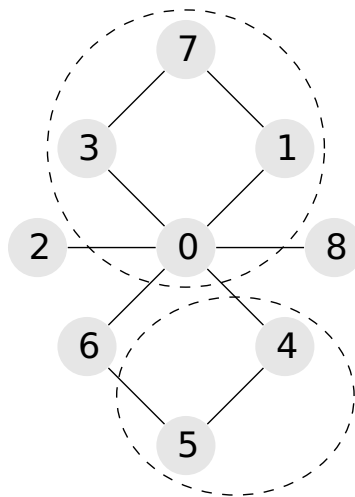
A , B ou C , respectivement.

Exemples

Exemple 1

Considérez l'appel de fonction suivant :

```
find_split(9, 4, 2, 3, [0, 0, 0, 0, 0, 0, 1, 3, 4, 5],  
           [1, 2, 3, 4, 6, 8, 7, 7, 5, 6])
```

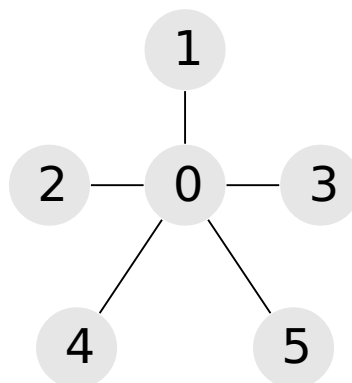


Une solution correcte possible est $[1, 1, 3, 1, 2, 2, 3, 1, 3]$. Cette solution correspond à la partition suivante : $A = \{0, 1, 3, 7\}$, $B = \{4, 5\}$ et $C = \{2, 6, 8\}$. Les ensembles A et B sont connexes.

Exemple 2

Considérez l'appel de fonction suivant :

```
find_split(6, 2, 2, 2, [0, 0, 0, 0, 0], [1, 2, 3, 4, 5])
```



Aucune partition valide n'existe. Par conséquent, la seule réponse correcte est

[0, 0, 0, 0, 0, 0].

Contraintes

- $3 \leq n \leq 100\,000$
- $2 \leq m \leq 200\,000$
- $1 \leq a, b, c \leq n$
- $a + b + c = n$
- Il y a au plus une route entre chaque paire d'attractions
- Il est possible de se déplacer entre n'importe quelle paire d'attractions en passant par les routes.
- $0 \leq p[i], q[i] \leq n - 1$ et $p[i] \neq q[i]$ pour $0 \leq i \leq m - 1$

Sous-tâches

1. (7 points) Chaque attraction est l'extrémité d'au plus deux routes.
2. (11 points) $a = 1$
3. (22 points) $m = n - 1$
4. (24 points) $n \leq 2500, m \leq 5000$
5. (36 points) Aucune contrainte supplémentaire.

Évaluateur d'exemple (sample grader)

L'évaluateur d'exemple lit l'entrée dans le format suivant :

- ligne 1: $n \ m$
- ligne 2: $a \ b \ c$
- ligne $3 + i$ (pour $0 \leq i \leq m - 1$): $p[i] \ q[i]$

L'évaluateur d'exemple affiche une unique ligne, contenant le tableau renvoyé par `find_split`.